



A Panchromatic Gamma Ray Burst MIDEX Mission

***ITOS Naming Convention/Database Format
Control Document (DFCD)***

Swift-OMI-013

**Version 1.2
May 30, 2002**

Swift Ground Segment
Omitron, Inc.

PREPARED BY: _____ DATE
Dustin Aldridge, Omitron Ground System Engineer

APPROVED BY: _____ DATE
Frank Marshall, GSFC Code 662, GNEST Lead

APPROVED BY: _____ DATE
Margaret Chester, PSU Operations Lead

APPROVED BY: _____ DATE
Lisa Nelson, Spectrum Astro Operations

APPROVED BY: _____ DATE
Pete Roming, PSU UVOT Instrument Lead

APPROVED BY: _____ DATE
Dave Burrows, PSU XRT Instrument Lead

APPROVED BY: _____ DATE
Jack Tueller, GSFC Code 661, BAT Instrument Lead

APPROVED BY: _____ DATE
John Ong, GSFC Code 582, Swift Software Systems Manager

APPROVED BY: _____ DATE
Doug Spiegel, Omitron Ground Segment Manager

REVISION SUMMARY

REV	RELEASE DATE	BRIEF DESCRIPTION/REASON FOR CHANGE	EFFECTIVE PAGES
D2	XX Month XXXX	Initial Draft 2 release by Spectrum Astro.	All
1.0	13 March 2001	Added more database formatting information	All
1.1	12 April 2001	Updated telecommand format definition	All
1.2	30 May 2002	Inserted PROC Style Guide	All

TABLE OF CONTENTS

	<u>PAGE</u>
1. INTRODUCTION	1
1.1 Purpose.....	1
1.2 Scope.....	1
2. APPLICABLE AND REFERENCE DOCUMENTS	1
2.1 Applicable Documents	1
2.1.1 Other Documents.....	1
3. DEFINITIONS.....	1
4. ITOS NAMING CONVENTIONS	1
4.1 General Naming Convention.....	1
4.2 General Mnemonic Naming Convention.....	2
4.3 Telecommands.....	2
4.4 Telemetry.....	2
4.5 Subsystem Names.....	3
4.6 ITOS Field Delimiters.....	3
4.7 Flight Paramaters.....	3
4.8 Filename Conventions.....	3
4.8.1 ITOS Display Pages	3
4.8.2 STOL Procs	3
4.8.2.1 STOL Test Procs.....	4
4.8.2.2 STOL Procedure Name Examples	4
4.8.3 Table Load Files.....	4
4.8.3.1 Flight Parameters	5
4.8.3.2 RTS Table Loads	5
4.8.3.2.1 RTS Activity Definitions.....	5
4.8.3.2.2 RTS Load File Names.....	5
4.8.3.3 ATS Table Loads	5
4.8.4 Database Exchange Files.....	5
4.8.4.1 Database Exchange File Filename Examples	5
4.8.4.2 Database Exchange File Format.....	6
5. SWIFT ITOS PROC STYLE GUIDE	7

APPENDIX A: PROCEDURE EXAMPLES.....	8
-------------------------------------	---

LIST OF TABLES

Table 4-1 Single-Character Prefix Assignment	1
Table 4-2 Telemetry Packet Header Telemetry Points	2

LIST OF ACRONYMS AND ABBREVIATIONS

APID	Application Process Identifier
ATS	Absolute Time Sequence
BAT	Burst Alert Telescope
FoM	Figure Of Merit
I&T	Integration and Test
ITOS	Integrated Test and Operations System
PFS	Product Function Specification
RTS	Relative Time Sequence
S/C	Spacecraft
UVOT DPU	Ultraviolet Optical Telescope Data Processing Unit
UVOT ICU	Ultraviolet Optical Telescope Instrument Control Unit
XRT	X-Ray Telescope

1. INTRODUCTION

1.1 Purpose

This document contains a set of guidelines describing the Swift Project naming conventions for the telecommand, telemetry, discretes, analogs, limits, and flight parameter mnemonics, and filenames used for the ITOS ground system. ITOS Proc style guidelines are also provided. Because the Swift program has multiple teams contributing to the Swift ITOS test and ground system, it is important that a naming convention is used to avoid conflicts.

1.2 Scope

These guidelines should be followed by all teams contributing to the Swift ITOS test and ground system.

2. APPLICABLE AND REFERENCE DOCUMENTS

2.1 Applicable Documents

Unless otherwise specified, the following documents in their current issue form a part of this document to the extent specified herein.

2.1.1 Other Documents

- a. <http://itos.gsfc.nasa.gov/> Integrated Test and Operations System (ITOS)

3. DEFINITIONS

N/A

4. ITOS NAMING CONVENTIONS

Because the Swift program has multiple different teams defining telecommand and telemetry mnemonics, it is important that a naming convention be used to avoid conflicts in assigned names. The following is the naming conventions that will be used for the telecommand, telemetry, discretes, analogs, limits, and flight parameter mnemonics, and filenames for the ITOS ground system.

4.1 General Naming Convention

In general, each ITOS database mnemonic or filename must begin with a single-character prefix assigned to each team as enumerated in Table 4-1.

Table 4-1 Single-Character Prefix Assignment

Prefix	Team Assignment
B	BAT
D	UVOT DPU
F	FoM
I	UVOT ICU
S	Spacecraft
U	UVOT (general)
X	XRT

4.2 General Mnemonic Naming Convention

In general, mnemonics have a maximum length of 16 characters (this is the limit imposed by the ITOS ground system). All mnemonics must begin with a pre-assigned single-character prefix. Because of the 16-character limitation for naming mnemonics, the use of underscores to delimit names is not required. Please review the ITOS database exchange record documentation to clarify field name definitions.

4.3 Telecommands

All telecommand mnemonics (“CMD” record type) have a maximum length of 16 characters, and must start with a pre-assigned single-character prefix.

For the ITOS “FLD” record type, the telecommand mnemonics and the “discrete value set names” have a maximum length of 16 characters and must start with a pre-assigned single-character prefix. The “field name” in the “FLD” record type does not require a prefix.

For the ITOS “SUB” record type, the “discrete value set names” have a maximum length of 16 characters, and must start with a pre-assigned single-character prefix. The “value name” field in the “SUB” record type does not require a prefix.

4.4 Telemetry

All telemetry mnemonics (“TLM” record type) have a maximum length of 16 characters, and must start with a pre-assigned single-character prefix.

All telemetry analog conversion definition names (“ALG” record type) have a maximum length of 16 characters, and must start with a pre-assigned single-character prefix.

All telemetry discrete conversion definition names (“DSC” record type) have a maximum length of 16 characters, and must start with a pre-assigned single-character prefix.

All telemetry limit definition names (“LIM” record type) have a maximum length of 16 characters, and must start with a pre-assigned single-character prefix.

An exception to the general naming convention is ITOS specified table and memory dump mnemonics. These mnemonics are defined by ITOS and are used to trigger dump collection.

Another exception to the general naming convention is for telemetry packet header field mnemonics, which will be standardized with the format of the single character ‘H’, followed by a four decimal-digit telemetry packet APID, followed by a field name. Table 4-2 contains the standardized fields for APID 123.

Table 4-2 Telemetry Packet Header Telemetry Points

Mnemonic	Start Byte	Start Bit	Length	Data Type	Description
H0123APID	0	0	11	U12	APID 123 Packet Application Process ID
H0123SHDRFLG	0	11	1	U12	APID 123 Packet Secondary Header Flag
H0123PKTTYPE	0	12	1	U12	APID 123 Packet Type
H0123PKTVNO	0	13	3	U12	APID 123 Packet Version Number
H0123SEOCNT	2	0	14	U12	APID 123 Packet Sequence Count
H0123SEGFLG	2	14	2	U12	APID 123 Packet Sequence Flags
H0123PKTLEN	4	0	16	U12	APID 123 Packet Length
H0123TIME	6	0	48	STIME42	APID 123 S/C Clock time when packet data was collected
H0123SECONDS	6	0	32	U1234	APID 123 S/C Clock seconds value when packet data was collected

H0123SUBSECS	10	0	16	U12	APID 123 S/C Clock subseconds value when packet data was collected
--------------	----	---	----	-----	--

4.5 Subsystem Names

All subsystem name mnemonics (“SSI” record type) have a maximum length of 16 characters, and must start with a pre-assigned single-character prefix.

4.6 ITOS Field Delimiters

The first record in each ITOS database exchange record file must be a field delimiter definition record (“DEL” record type). The Swift project will use the vertical bar (“|”) as the standard field delimiter. The following is how this record must be formatted:

DEL, |

4.7 Flight Parameters

Another exception to the general telemetry naming convention is for flight parameters. All flight parameters will start with the letter ‘V’ (for ‘V’ariable) followed by a pre-assigned single-character subsystem prefix.

4.8 Filename Conventions

All of the files used to define database exchange records, display pages, STOL scripts, etc., must have unique names. Filenames will consist of all lowercase characters. To avoid filename conflicts, each filename must start with a pre-assigned single-letter prefix. In addition, ITOS sometimes allows files to be located in multiple directories; when this capability is available, each team will locate their files in separate directories with the following name:

- a. sc – Spacecraft team
- b. bat – BAT and FoM team
- c. uvot – UVOT team
- d. xrt – XRT team
- e. it – I&T team
- f. ops – Operations team

When multiple directories are used, each team is free to use additional subdirectories under their assigned directory name.

4.8.1 ITOS Display Pages

ITOS display pages will reside in multiple team directories starting from the \$HOME/swift/pages directory (i.e. \$HOME/swift/pages/sc, \$HOME/swift/pages/bat, etc.). The ITOS mnemonic GBL_PAGEPATH will be set to reference each of the subdirectories. Each display page filename must start with a pre-assigned single-letter prefix.

4.8.2 STOL Procs

ITOS STOL procs will reside in multiple team directories starting from the \$HOME/swift/procs directory (i.e. \$HOME/swift/procs/sc, \$HOME/swift/procs/bat, etc.). The ITOS mnemonic GBL_PROCSPATH will be set to reference each of the subdirectories. Each STOL procedure filename must start with a pre-assigned single-letter prefix. The filename of the procedure must be the same as the procedure name with a ‘.proc’ suffix appended.

4.8.2.1 STOL Test Procs

As part of the procedure name, test procedures will have an addition three-character prefix appended after the pre-assigned single-character prefix. The following are the three letter prefixes that indicate the test type:

- a. cpt - Comprehensive Performance Test
Functional test which is designed to check every piece of hardware and every interface in the subsystem to the extent possible. This will be run as part of the full system functional test run occasionally during mission I&T.
- b. lpt - Limited Performance Test
A short-form functional, designed to take a less than four hours per subsystem, which is run more frequently.
- c. sft - Special Functional Tests
These are special calibration and other one-time tests outside of the other categories.

4.8.2.2 STOL Procedure Name Examples

The following are examples of STOL Proc names:

- a. scrcset
Filename: scrcset.proc. Send a S/C hardware command.
- b. spowerup
Filename: spowerup.proc. Power up the spacecraft.
- c. scpt_sm
Filename: scpt_sm.proc. The top level comprehensive test script for the Storage Management subsystem.
- d. bcpt_batnoop.proc
Filename: bcpt_batnoop.proc: The comprehensive test script for the BATNOOP telecommand.
- e. slpr_sm_verify
Filename: slpt_sm_verify.proc. The limited test script for the Storage Management subsystem that verifies all telemetry points for that subsystem.
- f. icpt_abcddev
Filename: icpt_abcddev.proc. Top level comprehensive test script for the ABC hardware device
- g. slpt_pcbaud_sep_cntrl
Filename: slpt_pcbaud_relay_cntrl.proc. The comprehensive test script for testing the PCB relays.

4.8.3 Table Load Files

ITOS table load files will reside in multiple team directories starting from the \$HOME/swift/loads directory (i.e. \$HOME/swift/loads/sc, \$HOME/swift/loads/bat, etc.). The ITOS mnemonic GBL_IMGLOaddir will be set to reference each of the subdirectories. Each table load file name must start with a pre-assigned single-letter prefix.

There are three files in this directory that must have a specific name; each team will locate each of these files in their team subdirectory:

- a. tablefields.data – file used by the ITOS leditor program to specify user defined table formats.
- b. rtsfiles.data - file is used by the ITOS leditor program to define a Relative Time Sequence (RTS) activity from which RTS table loads are generated.
- c. atsfiles.data - file is used by the ITOS leditor program to define a Absolute Time Sequence (ATS) activity from which ATS table loads are generated

4.8.3.1 Flight Parameters

The ‘tablefields.data’ file is used by the ITOS leditor program to specify user defined table formats.

4.8.3.2 RTS Table Loads

The ‘rtsfiles.data’ file is used by the ITOS leditor program to define RTS activities from which RTS table loads are generated.

4.8.3.2.1 RTS Activity Definitions

The leditor program defines an “activity” as a grouping of telecommands. The activity name will follow the following convention:

- a. by a single-letter pre-assigned prefix
- b. the three letter ‘rts’ prefix followed by the two digit RTS number. Example: rts05
- c. an underscore followed by a descriptive name.

Examples are: srt40_smnoop, xrt50_max_rts_starts

4.8.3.2.2 RTS Load File Names

When an RTS table load is generated, the resulting file will be named the same as the RTS activity name, with an ending file suffix of “.atf”.

Examples are: srt40_smnoop.atf, xrt50_max_rts_starts.atf

4.8.3.3 ATS Table Loads

The ‘atsfiles.data’ file is used by the ITOS leditor program to define ATS activities from which ATS table loads are generated. The naming conventions will follow those used for RTS Table Loads, with the exception that the prefix ‘rts’ is replaced by ‘ats’.

4.8.4 Database Exchange Files

ITOS database exchange files will reside in multiple team directories starting from the \$HOME/swift/dbx directory (i.e. \$HOME/swift/dbx/sc, \$HOME/swift/dbx/bat, etc.). Each database exchange file filename must start with a pre-assigned single-letter prefix. Each database exchange file filename must have a ‘.dbx’ suffix.

4.8.4.1 Database Exchange File Filename Examples

- a. BAT detector array plate telemetry
Filename: bdat_tlm.dbx
- b. BAT detector array plate commands
Filename: bdat_cmd.dbx
- c. BAT command data handler telemetry
Filename: bcdh_tlm.dbx
- d. BAT command data handler commands
Filename: bcdh_cmd.dbx
- e. Spacecraft C&DH telemetry
Filename: scdh_tlm.dbx
- f. Spacecraft C&DH commands

Filename: scdh_cmd.dbx

4.8.4.2 Database Exchange File Format

ITOS formatting requirements already restrict the database exchange file format. Several additional recommendations will greatly improve the human readability of the dbx files:

- a. The vertical bar (“|”) will be used as the standard field delimiter as indicated in section 4.6.
- b. Within each record type, fixed column widths will be used for each field to improve readability.
- c. Comments and blank lines will be used liberally to distinguish apid sets, commands with multiple “FLD” records, and configuration control information.
- d. In telemetry definitions, the “TLM” and “PKT” records will be specified in the same byte order as the data is organized in the telemetry packet.
- e. The “TLM”, “PKT”, “DSC”, “ALG”, and “LIM” records for a given telemetry mnemonic will all be contained in the same file unless the same “DSC”, “ALG”, or “LIM” records for that mnemonic are used by several subsystems.
- f. In telecommand definitions, the “FLD” records for a given telecommand will immediately follow the “CMD” records for that telecommand.

Examples of an implementation of these database recommendations are located on the GNEST web site.

5. Swift ITOS Proc Style Guide

The following style guidelines for ITOS procs are recommended for maximum commonality and reuse between spacecraft bus I&T, instrument I&T, observatory I&T, and operations:

1. Proc headers will conform to proc_header.proc in Appendix A. The proc header will include at a minimum: mission name, proc title, author, creation date, purpose, responsible lead, and a revision history with any necessary automated revision history tags.
2. All ITOS directives will be in upper case.
3. All local and global variables will be in lower case.
4. Any command, telemetry, or system mnemonics or submnemonics will be in uppercase.
5. All labels will be in upper case. All labels will be called in uppercase.
6. The indentation for all loops and IF statements will be four spaces.
7. No tabs will be used.
8. Blank lines will be allowed but a comment ";" is encouraged instead for readability.
9. Statements in ASK and WRITE statements as well as comments will be in upper and lower case to improve readability.
10. Nesting will be limited to five levels.
11. Structured programming is encouraged and GOTOs will be limited.
12. All procs will end with the following two lines:

```
SHO "PROC [PROCNAME] COMPLETED."  
ENDPROC
```

13. All commanding will be in "ONESTEP" mode.
14. All relational operators like ".EQ.", ".AND.", and ".OR." will be in uppercase and have spaces around them to improve readability as in the following example:

```
WAIT UNTIL (MNEMONIC1 .EQ. (MOD((localex + 1),256)))
```

Extremely critical for PROCS destined for operations reuse:

15. All procs destined for operations reuse will contain command end item telemetry verification using "WAIT UNTIL" statements.
16. All variables passed to the procedure will be verified whenever possible.
17. All ASK statements requiring discrete input or known ranges will be structured in a DO WHILE loop that will check for correct responses.

APPENDIX A: PROCEDURE EXAMPLES

```
PROC PROC_HEADER
;-----
; MISSION: SWIFT
; TITLE:
; AUTHOR:
; DATE:
; PURPOSE:
;-----
; RESPONSIBLE LEAD:
;-----
; PROCEDURE CALL OUTS:
; NAME:           ACTIVITY:
; -----          -----
; 
;-----
; REVISION HISTORY:
; $Id: proc_header.proc,v 1.2 2000/09/26 14:09:16 swift Exp $
; DATE:    INITIALS:    CHANGES:
; -----    -----    -----
; 
; 
; ****
; -----
;----- DEFINE PROCEDURE VARIABLES -----
GLOBAL
LOCAL

SHO "PROCEDURE PROC_HEADER COMPLETE."
ENDPROC
```

```

PROC NIFIXOPEN
;-----
;
;   MISSION: Triana
;
;   TITLE:   NISTAR Set Fixed Open Precharge
;
;   AUTHOR:  Joseph Kowalski, Honeywell Technology Solutions Inc.
;
;   DATE:    April 21, 2000
;
;   PURPOSE: This procedure sets the fixed open precharge for the
;             drop floor algorithm. It should only be run when NISTAR
;             is in Science Mode.
;
;-----
;
;   RESPONSIBLE LEAD: Dr. Steven Lorentz, NISTAR PI
;
;-----
;
;   PROCEDURE CALL OUTS:
;
;       NAME:      ACTIVITY:
;       -----  -----
;
;
;-----
;
;   REVISION HISTORY:
;
;       $Id: nifixopen.proc,v 1.4 2000/09/18 19:34:31 triana Exp $
;
;       DATE:      INITIALS:      CHANGES:
;       -----  -----
;       5/15/00     MRR          Changed command after database change
;       7/17/00     MRR          Changed incorrect command field
;       9/18/00     JJK          Added ask statement to begin procedure
;
;
;
;*****----- DEFINE PROCEDURE VARIABLES -----*
;
;----- LOCAL cmdcnt, cmderrcnt, device, filter, setpt
;
; IF (P@NIINSTMODE .NE. "SCIENCE") THEN
;     SHO "Instrument not in Science mode"
;     GOTO END
ENDIF
;
ASK "Hit 'OK' to configure fixed open precharge"
;
cmdcnt = NICMDCNT ; Command Accept Counter
cmderrcnt = NIERRCNT ; Command Reject Counter
;
device = 1 ; Initialize variable
filter = 1 ; Initialize variable
setpt = "x" ; Initialize variable
;
DO UNTIL ((device .EQ. "RC1") .OR. (device .EQ. "RC2") .OR. (device .EQ. "RC3"))
    ASK "Select the device (RC1/RC2/RC3)", device
    device = UPPERCASE(device)
ENDDO
;
DO UNTIL ((filter .EQ. "A") .OR. (filter .EQ. "B") .OR. (filter .EQ. "C"))
    ASK "Select the filter (A/B/C)", filter
    filter = UPPERCASE(filter)
ENDDO
;
ASK "Enter the setpoint for the precharge", setpt

```

```

;
/NIFOPRECHRG DEVICE=name(device), FILTER=name(filter), SETPOINT=(setpt)
;
WAIT UNTIL (NICMDCNT .EQ. MOD((cmdcnt + 1),256) .AND. (NIERRCNT .EQ. cmderrcnt))
IF ((device .EQ. "RC1") .AND. (filter .EQ. "A")) THEN
    WAIT UNTIL (NIRC1FOPRECHRGA .EQ. setpt)
ELSEIF ((device .EQ. "RC1") .AND. (filter .EQ. "B")) THEN
    WAIT UNTIL (NIRC1FOPRECHRGB .EQ. setpt)
ELSEIF ((device .EQ. "RC1") .AND. (filter .EQ. "C")) THEN
    WAIT UNTIL (NIRC1FOPRECHRCG .EQ. setpt)
ELSEIF ((device .EQ. "RC2") .AND. (filter .EQ. "A")) THEN
    WAIT UNTIL (NIRC2FOPRECHRGA .EQ. setpt)
ELSEIF ((device .EQ. "RC2") .AND. (filter .EQ. "B")) THEN
    WAIT UNTIL (NIRC2FOPRECHRGB .EQ. setpt)
ELSEIF ((device .EQ. "RC2") .AND. (filter .EQ. "C")) THEN
    WAIT UNTIL (NIRC2FOPRECHRCG .EQ. setpt)
ELSEIF ((device .EQ. "RC3") .AND. (filter .EQ. "A")) THEN
    WAIT UNTIL (NIRC3FOPRECHRGA .EQ. setpt)
ELSEIF ((device .EQ. "RC3") .AND. (filter .EQ. "B")) THEN
    WAIT UNTIL (NIRC3FOPRECHRGB .EQ. setpt)
ELSEIF ((device .EQ. "RC3") .AND. (filter .EQ. "C")) THEN
    WAIT UNTIL (NIRC3FOPRECHRCG .EQ. setpt)
ENDIF
;
END:
SHO "PROCEDURE NIFIXOPEN COMPLETED."
ENDPROC

```

```

PROC TCATSSTART
;-----
;
; MISSION: Example
;
; TITLE: Start ATS
;
; AUTHOR: Test Conductor
;
; DATE: 10/20/2000
;
; PURPOSE: Start ATS buffer
;
;-----
;
; RESPONSIBLE LEAD: C&DH Lead
;
;-----
;
; PROCEDURE CALL OUTS:
;
; NAME: ACTIVITY:
; ----- -----
;
;
;-----
;
; REVISION HISTORY:
;
; $Id: proc_header.proc,v 1.2 2000/09/26 14:09:16 swift Exp $
;
; DATE: INITIALS: CHANGES:
; ----- ----- -----
; 10/22/2000 DJA Fixed bug in initial ask statement.
;
;
;*****
;----- DEFINE PROCEDURE VARIABLES -----
LOCAL ans, answer, cmdctr
;
; START THE ATS
;
; SELECT WHICH ATS BUFFER TO START, A OR B
;
ans="x"
DO WHILE ((ans .NE. "A") .AND. (ans .NE. "B") .AND. (ans .NE. "N"))
    ASK "Start ATS in buffer A OR B (A/B) OR N = NO to starting a buffer", ans
    ans = UPPERCASE(ans)
ENDDO
;
IF ((ans .EQ. "A") .AND. (SCATPSTATE .NE. 2)) THEN
;
    cmdctr = SCCMDCNT
    /SCATSSTART BUFA
;
    WAIT UNTIL (SCCMDCNT .EQ. MOD((cmdctr + 1),256))
    WAIT UNTIL (SCATSCURNUM .EQ. 1) ; Verify buf A
    WAIT UNTIL (SCATPSTATE .EQ. 2) ; Verify execution
;
ELSEIF ((ans .EQ. "A") .AND. (SCATPSTATE .EQ. 2)) THEN
;
    ASK "Buffer A is currently executing a load. Hit 'OK' to kill proc!"
    GOTO STOPPROC
;
ELSEIF ((ans .EQ. "B") .AND. (SCATPSTATE .NE. 2)) THEN
;
    cmdctr = SCCMDCNT
    /SCATSSTART BUFB

```

```

;
WAIT UNTIL (SCCMDCNT .EQ. MOD((cmdctr + 1),256))
WAIT UNTIL (SCATSCURNUM .EQ. 2) ; Verify buf B
WAIT UNTIL (SCATPSTATE .EQ. 2) ; Verify Execution
;
ELSEIF ((ans .eq."B") .and. (SCATPSTATE .EQ. 2)) THEN
;
ASK "Buffer B is currently executing a load. Hit 'OK' to kill proc!"
GOTO STOPPROC
;
ELSE
    GOTO OUT
ENDIF
;
GOTO OUT
;
STOPPROC:
;
;
REM; To stop the current ATS from executing and then start the new ATS,
REM; Perform the following steps at the STOL prompt:
REM;      1. Type /SCATSSTOP
REM;      2. Type START TCATPSTART
WAIT;
;
OUT:
END:
SHOVAL "PROCEDURE TCATSSTART IS COMPLETE"
ENDPROC

```

```

PROC SCPWRDOWN
;-----
;
; MISSION: Triana
;
; TITLE: Triana Spacecraft Power Down Procedure
;
; AUTHOR: Rick Saylor, AlliedSignal Technical Services Corporation
;
; DATE: January 24, 2000
;
; PURPOSE: This procedure will power down the spacecraft
;
;-----
;
; RESPONSIBLE LEAD: Tim Trenkle, Triana Lead System Engineer
;
;-----
;
; PROCEDURE CALL OUTS:
;
; NAME: ACTIVITY:
; ----- -----
; status_ddr Verifies the DDR recording status
; totlmon Turns on telemetry at specified rate
; egsepwroff Disables the output on the EGSE power supplies
; stop_ddr Stops recording on the DDR
;
;-----
;
; REVISION HISTORY:
;
; $Id: scpwrdown.proc,v 1.26 2000/07/20 19:22:24 triana Exp $
;
; DATE: INITIALS: CHANGES:
; ----- -----
; 2/16/00 RS - Changed time jam to wallclock
; 2/17/00 RS - Modified procedure, commented items that
;               doesn't apply yet.
; 2/28/00 RS - Added ground mnemonics to proc.
; 3/14/00 DL - Added check to see if EGSE rack was powered down
;               and a page close of the instructions.
; 4/4/00 MRR - Removed command to boot mode before power off
;               added commented out call to rf_cmdoff at end.
; 4/20/00 DL - Added call to page for UART Shutdown
; 4/20/00 DL - Change the location of the UART page call
; 5/31/00 MRR - Changed mnemonic names for RWs.
; 6/08/00 JJK - Added ask to turn off 5V pwr sup on top of tcw rack
; 6/08/00 MRR - Added power node laptop shutdown and strip charting shutdown
; 6/12/00 MRR - Added option to turn off SAS Rack
; 6/12/00 MRR - Uncommented timeout seqprt clear
; 6/13/00 RS - Updated PN relay commands
; 6/20/00 MRR - Removed references to 5V power supply since CDH SW load
;               fixes problem
; 6/20/00 RS - Fixed bugs in procedure
; 6/20/00 MRR - Fixed bugs in procedure
; 6/22/00 RS - Fixed and updated wheel section for 2 ETU wheels
; 6/22/00 RS - Updated log info at end of proc.
; 7/20/00 RS - Added battery offline section
; 7/20/00 RS - Changed strip chart machine to LA-NINA
; 7/20/00 RS - Added wheel commands/verification
;
;*****
;
;----- DEFINE PROCEDURE VARIABLES -----
;
LOCAL ans1, ans2, x1, x2
LOCAL log_info      ; Holds log info for power down log
LOCAL ptkcnt        ; Packet counter
LOCAL time          ; Time tag for log information
LOCAL cmd_cnt       ; Command counter
LOCAL cmd_err       ; Command error counter
LOCAL pul_cnt       ; Pulse counter
;
;
;----- INITIALIZED VARIABLES -----
;
ans1 = "X"
ans2 = "X"
x1 = "X"
x2 = "X"

```

```

;
;----- CHECK FOR ITOS LOG STATUS -----
;
IF (gbl_logevent .NE. "ON") THEN
;
; Start the daily event log
LOG >>
(CONCAT("EVENT_",SUBSTR(p@GBL_WALLCLOCK,1,2),"_",SUBSTR(p@GBL_WALLCLOCK,4,3),"_",SUBSTR(p@GBL_WALLCLOCK,8,2),SU
BSTR(p@GBL_WALLCLOCK,11,2),".LOG") )
;
ENDIF
;
;----- VERIFY DDR STATUS -----
;
START STATUS_DDR ; Verify DDR Status
;
;----- TURN OFF CONFIGMONS -----
;
CFGMON CLEAR ALL
;
CFGMON1="OFF"
CFGMON2="OFF"
CFGMON3="OFF"
CFGMON4="OFF"
CFGMON5="OFF"
CFGMONUHUB="OFF"
;
;----- ITOS/SCAT CHECKS -----
;
ITOSCONFIG:
;
IF (UPPERCASE(GBL_TM_CONNECT) .EQ. "DISCONNECTED") THEN
;
ENABLE TLM ; Enable ITOS telemetry processor
WAIT 2
;
AC NAME(TO_TLM_HOST)
;
WAIT UNTIL (UPPERCASE(GBL_TM_CONNECT) .EQ. "CONNECTED"))
WAIT UNTIL (UPPERCASE(GBL_ACQUIRE) .NE. "OFF")
WAIT 2
;
SHO "Telemetry connection made to SCAT"
;
ENDIF
;
IF (GBL_CM_CONNECT .EQ. "DISCONNECTED") THEN
;
ENABLE CMD
WAIT UNTIL (GBL_CM_CONNECT .EQ. "CONNECTED")
SHO "Command Connection made to SCAT"
;
ENDIF
;
/MODE BYPASS OFF
/MODE STEP1
/MODE REXMIT OFF
;
WAIT UNTIL (GBL_BP_MODE .EQ. 0)
WAIT UNTIL (GBL_CMD_MODE .EQ. 3)
WAIT UNTIL (GBL_RETRLIM_0 .EQ. 0)
;
pktcnt = H001CNT
;
WAIT UNTIL (H001CNT .NE. pktcnt) ; Check for telemetry update
;
;----- S/C TIME UPDATE -----
;
TIMEJAM:
;
;ASK "VERIFY THAT THE TIME CODE GENERATOR IS SET TO GMT OR DESIRED SETTING"
;
;DATE SYNC ; Grab the GMT / Simulated Time
;
;WAIT 2
;
;cmd_cnt = SHTCCMDCNT
;cmd_err = SHTCERRCNT
;
;/TCTIMEJAM GMT=p@GBL_WALLCLOCK ; Jam Spacecraft Time
;

```

```

;WAIT UNTIL (SHTCCMDCNT .EQ. (MOD((cmd_cnt + 1), 256)))
;WAIT UNTIL (SHTCERRCNT .EQ. cmd_err)
;
;WAIT UNTIL (HKEVENTCODE .EQ. 50) ; Verify clock adjust event was generated
;
;----- BULK MEMORY RECORDER DUMP -----
;
; TBD
; This section will dump the spacecraft recorder, ACS, ENG & Events
;
;----- VERIFY IRWA STATUS -----
;
; ((p@ARW1PWRST .EQ. "OFF") .OR. (p@ARW2PWRST .EQ. "OFF") .OR. (p@ARW3PWRST .EQ. "OFF") .OR. (p@ARW4PWRST .EQ.
;"OFF")) THEN
;
/AWHLPWR1 ON
/AWHLPWR2 ON
/AWHLPWR3 ON
/AWHLPWR4 ON
;
WAIT UNTIL (p@ARW1PWRST .EQ. "ON")
WAIT UNTIL (p@ARW2PWRST .EQ. "ON")
WAIT UNTIL (p@ARW3PWRST .EQ. "ON")
WAIT UNTIL (p@ARW4PWRST .EQ. "ON")
;
ELSE
;
SHO "WHEELS ARE ALREADY POWERED ON"
;
ENDIF
;
SHO "SPIN DOWN ALL WHEELS TO 0 r/s"
;
/AOVSPD DWELL=65000, RATE1=0, RATE2=0, RATE3=0, RATE4=0
;
WAIT UNTIL ((ACRWSPEED1 .LT. 10.0) .AND. (ACRWSPEED1 .GT. -10.0))
WAIT UNTIL ((ACRWSPEED2 .LT. 10.0) .AND. (ACRWSPEED2 .GT. -10.0))
WAIT UNTIL ((ACRWSPEED3 .LT. 10.0) .AND. (ACRWSPEED3 .GT. -10.0))
WAIT UNTIL ((ACRWSPEED4 .LT. 10.0) .AND. (ACRWSPEED4 .GT. -10.0))
;
/AWHLPOWR1 OFF
/AWHLPOWR2 OFF
/AWHLPOWR3 OFF
/AWHLPOWR4 OFF
;
WAIT UNTIL (p@ARW1PWRST .EQ. "OFF")
WAIT UNTIL (p@ARW2PWRST .EQ. "OFF")
WAIT UNTIL (p@ARW3PWRST .EQ. "OFF")
WAIT UNTIL (p@ARW4PWRST .EQ. "OFF")
;
SHO "ALL WHEELS ARE POWERED OFF"
;
;----- ACS SIMULATOR SECTION -----
;
; TBD
;
;----- RETURN TO HARLINE COMMANDING -----
;
;
IF (cmdmode .EQ. "RF") START RF_CMDOFF
;
IF (tlmmod .EQ. "RF") START RF_TLMOFF
;
;
;----- BATTERY OFF-LINE -----
;
IF (batmode .EQ. "SIM") THEN
;
START PBATSIMOFF ; Disable battery simulator
;
ELSEIF (batmode .EQ. "BAT") THEN
;
START PBATOFF ; Disable flight/Qual. battery
;
ENDIF
;
;
;----- INSTRUMENT PRE TURN OFF -----
;

```

```

; TBD
; Will configure instrument for turn off
;
;----- SPACECRAFT POWER OFF -----
;
WAIT 1
;
/MODE BYPASS ON
;
/TOXMITPWR OFF
;
WAIT UNTIL (p@UHXMITERST .EQ. "OFF")
;
/PNNISTARPWR DISABLE
WAIT UNTIL (p@PNNISTARPWR .EQ. "OFF")
;
/PNEPICPWR DISABLE
WAIT UNTIL (p@PNEPICPWR .EQ. "OFF")
;
/PNPMHVON DISABLE
WAIT UNTIL (p@PNPMHVON .EQ. "OFF")
;
/PNPMHOFF ENABLE
WAIT UNTIL (p@PNPMHOFF .EQ. "OFF")
;
pul_cnt = PNPMHPULSECNT
/PNPMHPULSE
;
WAIT UNTIL (PNPMHPULSECNT .EQ. pul_cnt + 1)
;
/PNPMHOFF DISABLE
WAIT UNTIL (p@PNPMHOFF .EQ. "OFF")
;
/PNPMPHAPWR DISABLE
WAIT UNTIL (p@PNPMPHAPWR .EQ. "OFF")
;
/PNHTR1PWR DISABLE
/PNHTR1PWR DISABLE
WAIT UNTIL (p@PNHTR1PWR .EQ. "OFF")
WAIT UNTIL (p@PNHTR1PWR .EQ. "OFF")
;
WAIT 2
;
IF (sasstat .EQ. "ON") THEN
    START SASOFF      ; Disable SAS Power supplies
ELSE
    START EGSEPWROFF   ; DISABLE EGSE POWER SUPPLIES
ENDIF
;
sasstat = "OFF"
;
;
ASK "HIT 'OK' AFTER VERIFYING S/C POWER IS OFF"
;
START STOP_DDR      ; Stop DDR recording
;
;----- POWER NODE LAPTOP POWER DOWN -----
;----- AND STRIP CHART TERMINATION -----
;
OPEN (50) LA-NINA:45000
WRITE (50) "START STRIP_CHARTOFF"
CLOSE (50)
;
ASK "QUIT POWER NODE MONITOR SOFTWARE AND SHUTDOWN LAPTOP"
;
;
;
;----- UART PWR DOWN -----
;
PAGE UART_SHUTDOWN   ; Steps for power down the EGSE rack
;
ASK "HIT 'OK' AFTER UART IS POWERED OFF"
;
PAGE CLEAR UART_SHUTDOWN
;----- EGSE PWR DOWN -----
;
PAGE EGSE_SHUTDOWN   ; Steps for power down the EGSE rack
;
ASK "HIT 'OK' AFTER EGSE RACK IS POWERED OFF"
;
PAGE CLEAR EGSE_SHUTDOWN

```

```

;
;
ASK "Power off all SAS power supplies and turn off SAS rack in back"
;
----- ARCHIVE STATUS -----
;
ARCHIVEST:
;
ARCHIVE STOP ALL      ; Stops all telemetry archiving
;
ARCHIVEST = "OFF"
;
LOG STOP              ; Stops log event recording
;
LOGSTATUS = "OFF"
;
archive_status = "OFF"
gbl_logevent = "OFF"
;
----- TIMEON SEQPRT -----
;
SEQPRT CLEAR TIMEON   ; Turns off timeon seqprt
;
----- LOG INFORMATION -----
;
START weeklylog        ; Start weeklylog procedure to calculate timeon data
;
;
time = SUBSTR(p@GBL_WALLCLOCK, 1, 15)
;
log_info = CONCAT(time, ", SCPWRDOWN")
;
OPEN (5) "/home/triana/SC_LOGS/scpwrdown.log"
WRITE (5) log_info
CLOSE (5)
;
----- DISABLE ITOS CONFIGURATION -----
;
DISABLE TLM           ; Disables telemetry controller
;
WAIT 1
;
-----
;
END:
SHO "PROCEDURE SCPWRDOWN COMPLETE."
ENDPROC

```